GetTempFileName

Use randomly generated prefix value to ensure filename that is more difficult to guess

Sean Barnum, Cigital, Inc. [vita[1]]

2007-03-23

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 7397 bytes

Attack Category	• File Manipulation • Encryption Assault • Path spoofing or confusion problem
Vulnerability Category	• Temporary file creation problem • Access Control
Software Context	• File Path Management
Location	• winbase.h
Description	When using GetTempFileName() to create a secure temporary file, care must be used to ensure that the name cannot be guessed. The GetTempFileName() function creates a name for a temporary file. If a unique file name is generated, an empty file is created and the handle to it is released; otherwise, only a file name is generated. The prefix should be a randomly generated value to ensure that an attacker cannot guess the name of a secure temporary file. When you need a secure temporary file, make sure that the temp file generation algorithm creates a unique and difficult-to-guess name. Also, ensure that the created file doesn't already exist and has appropriate access control permissions to protect against attackers. The last parameter, lpTempFileName, must be at least MAX_PATH characters in length or a buffer overflow could occur.
APIs	**Function Name** / **Comments** GetTempFileName GetTempFileNameA GetTempFileNameW
Method of Attack	An attacker could gain access to data in a temporary file by guessing the name of the file and creating it with permissions that allow the attacker access.

[1] http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html (Barnum, Sean)

<table>
<tr><td><b>Exception Criteria</b></td><td></td></tr>
<tr><td><b>Solutions</b></td><td>
<table>
<tr><th>Solution Applicability</th><th>Solution Description</th><th>Solution Efficacy</th></tr>
<tr><td>When a secure temporary file is needed.</td><td>Use CryptoAPI's CryptGenRandom to generate a random value for Prefix parameter.
Use the return value from GetTempPath() for the lpPathName parameter to ensure that the temp file is created in the Windows TEMP directory. This should ensure that the temporary file is created in a known, controlled locations.
Use CreateFile() to create the temporary file and specify CREATE_NEW for the dwCreationDisposition parameter. Check the return code for ERROR_EXISTS to guard against TOCTOU attacks. If the return value is ERROR_EXISTS, generate a different temp filename.
You should specify</td><td>Effective, depending on details.</td></tr>
</table>
</td></tr>
</table>

		FILE_ATTRIBUTE_NOT_CONTENT_INDEX FILE_ATTRIBUTE_TEMPORARY, and FILE_FLAG_DELETE_ON_CLOSE in the dwFlagsAndAttributes parameter of CreateFile() to provide additional performance and protection of temporary files. Example code for calling CreateFile to create a temporary file: HANDLE hTempFile = CreateFile(szTempFileName, // Temporary File Name GENERIC_READ | GENERIC_WRITE, // Desired Access 0, // Share Mode = NONE NULL, // Security Attributes = Use ACLs for TEMP directory CREATE_NEW, // Fails if file already exists FILE_ATTRIBUTE_NOT_CONTENT_INDEX | FILE_ATTRIBUTE_TEMPORARY | FILE_FLAG_DELETE_ON_CLOSE, NULL); // No Template File if (hTempFile == INVALID_HANDLE_VALUE) { // Error ! Count not create the file if (GetLastError()	

	==
	ERROR_FILE_EXISTS)
	{ // The temp file already exists! Generate another name and try again!
	}
	}

Signature Details	UINT GetTempFileName(LPCTSTR lpPathName, LPCTSTR lpPrefixString, UINT uUnique, LPTSTR lpTempFileName);

Examples of Incorrect Code	const DWORD BUFSIZE=MAX_PATH;
	char lpPathBuffer[BUFSIZE];
	if (!
	GetTempFileName(lpPathBuffer, // directory for temp files
	"NEW", // temp file name prefix
	0, // create unique name
	szTempName)) // buffer for name
	{
	handleError();
	}

Examples of Corrected Code	const DWORD BUFSIZE=MAX_PATH;
	const DWORD PREFIXSIZE=32;
	char lpPathBuffer[BUFSIZE];
	UINT uUnique;
	BYTE prefix [3]; //Because GetTempFileName only uses the first three
	if (!CryptGenRandom(hProv, 3, prefix) //Generate three random bytes
	return false; //Handle the error condition
	if (!
	GetTempFileName(lpPathBuffer, // directory for temp files
	(LPCTSTR *) prefix, // temp file name prefix, cast from bytes to a string
	0, // create unique name
	szTempName)) // buffer for name
	{
	handleError();
	}

<table>
<tr><td><b>Source Reference</b></td><td><ul><li>Howard, Michael & LeBlanc, David C. <i>Writing Secure Code, 1st ed</i>. Redmond, WA: Microsoft Press, 2002, ISBN: 0735615888. Chapter 16, “General Good Practices,” pp. 423-425, WSC1.</li></ul></td></tr>
<tr><td><b>Recommended Resources</b></td><td><ul><li>MSDN reference for GetTempFileName[2]</li><li>CryptGenRandom from MSDN[3]</li></ul></td></tr>
<tr><td><b>Discriminant Set</b></td><td><table><tr><td><b>Operating System</b></td><td><ul><li>Windows</li></ul></td></tr><tr><td><b>Languages</b></td><td><ul><li>C</li><li>C++</li></ul></td></tr></table></td></tr>
</table>

Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

[1] mailto:copyright@cigital.com